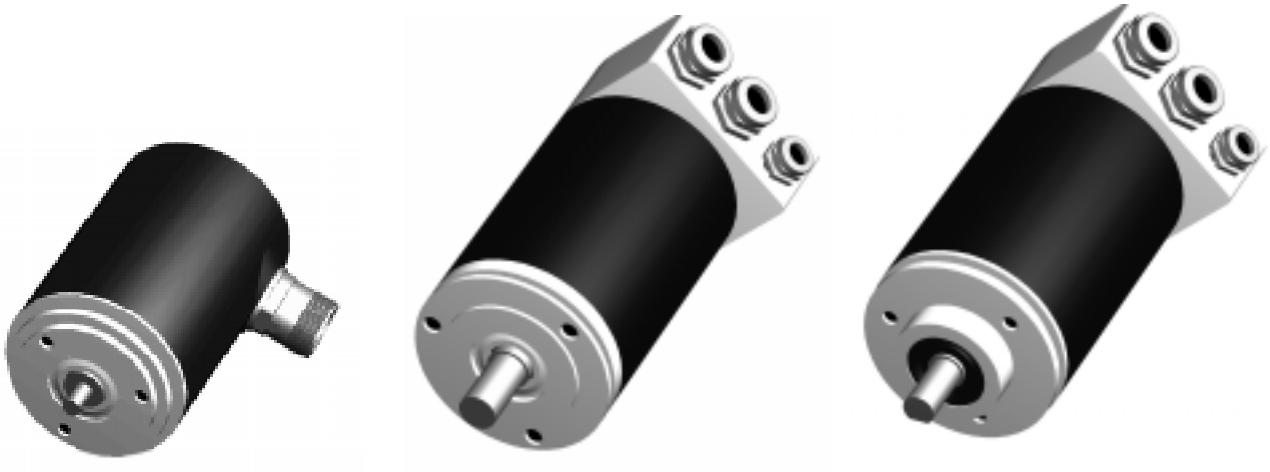
Operating Instructions



Page

0. Table of contents ..... 2 - 4

1. General 5

2. CANopen Communication Concept ..... 5 - 6

3. CANopen Scope of the TWK Encoder 6

4. Network Management ..... 7 - 8

4.1 Minimum-Capability-Device Boot up 7

4.2 Description of the NMT Commands 8

5. Node Guarding ..... 8 - 9

6. Bus Synchronisation 10

7. Emergency Message-supported Error Codes 10

8. Installation Instructions ..... 11 - 12

8.1 Node - ID, Baudrate, Terminal Resistor 11

8.2 LED Description 12

9. The Object Directory ..... 12 - 15

9.1 Subdivision of the Object Directory 12

9.2 Services for Writing and Reading the Object Directory 12

9.3 COB-ID structure 13

9.4 Message Objects and Function Codes 13

9.5 The Command Byte describes the Type of SDO Message 14

9.6 Examples of Service Data Objects 15

10. Communication Profile ..... 16 - 34

10.1 Object 1000H: Device Type ..... 16 - 17

10.2 Object 1001H: Error Register 17

10.3 Object 1002H: Manufacturer Status Register 18

10.4 Object 1003H: Predefined Error Field 19

10.5 Object 1004H: Number of PDOs supported 20

10.6 Object 1005H: COB-ID SYNC-Message 21

10.7 Object 1006H: Communication Cycle Period 22

10.8 Object 1007H: Synchronous Window Length 22
10.9 Object 1008H: Device Name 23
10.10 Object 1009H: Manufacturer Hardware Version 23
10.11 Object 100AH: Manufacturer Software Version 24
10.12 Object 100BH: Node-ID 24
10.13 Object 100CH: Guard Time 25
10.14 Object 100DH: Life Time Factor 25
10.15 Object 100EH: Node Guarding Identifier 26
10.16 Object 100FH: Number of SDO´s supported ..... 26 - 27
10.17 Object 1010H: Store Parameters 27
10.18 Object 1011H: Restore Default Parameters 28
10.19 Object 1014H: COB-ID Emergency Message 29
10.20 Overview of Transmission Types 29
10.21 Object 1800H: 1st Transmit PDO (asynchronous transmission) ..... 30 - 31
10.22 Object 1A00H: 1st Transmit PDO (asynchronous transmission) Mapping 32
10.23 Object 1802H: 2nd Transmit PDO (synchronous transmission) 33
10.24 Object 1A02H: 2nd Transmit PDO (asynchronous transmission) Mapping 34

11. Encoder Profile 35

11.1 Encoder Parameter 35
11.1.1 Object 6000H: Operating Parameters ..... 35 - 36
11.1.2 Object 6001H: Measuring Units per Revolution 36
11.1.3 Object 6002H: Total Measuring Range 37
11.1.4 Object 6003H: Preset Value 38
11.1.5 Object 6004H: Position Value ..... 38 - 39
11.1.6 Object 6200H: Cyclic Timer 39
11.1.7 Object 6400H: Area State Register 40
11.1.8 Object 6401H: Work Area low limit 41

11.1.9 Object 6402H: Work Area high limit 42

11.2. Encoder Diagnostics 43

11.2.1 Object 6500H: Operating Status 43

11.2.2 Object 6501H: Single Turn Resolution 44

11.2.3 Object 6502H: Number of distinguishable Revolutions ..... 44 - 45

11.2.4 Object 6503H: Alarms 45

11.2.5 Object 6504H: Supported Alarms 45

11.2.6 Object 6505H: Warnings 46

11.2.7 Object 6506H: Supported Warnings 46

11.2.8 Object 6507H: Profile and Software Version 47

11.2.9 Object 6508H: Operating Time 47

11.2.10 Object 6509H: Offset Value 48

11.2.11 Object 650AH: Module Identification 49

11.2.12 Object 650BH: Serial Number 50

11.3 Manufacturer-specific Profile Area 50

11.3.1 Object 2000H: Node ID (Plug Version) 51

11.3.2 Object 2001H: Baudrate (Plug Version) 52

12. Default Values for Programming Parameters ..... 52 - 54

13. Store Parameters (Object 1010H) 54

14. Literature 54

TWK

COPYRIGHT: The Operating Instructions CRN/C 10750
is owned by TWK-ELEKTRONIK GMBH and is
protected by copyright laws and international treaty provisions.

© 1999 by TWK-ELEKTRONIK GMBH
POB 10 50 63 ■ 40041 Düsseldorf ■ Germany
Tel. +49/211/63 20 67 ■ Fax +49/211/63 77 05
e-mail: info@twk.de ■ http://www.twk.de

1. General

The use of the CAN philosophy in industrial automation has led to the necessity of equipping encoders with this interface.

The basic prerequisites for the rapid expansion of the CAN-bus in industrial use have been the positive examples of application within the automobile industry, and the availability of inexpensive ICs from various manufacturers.

The CANopen profile has been developed on the basis of the CAN Application Layer (CAL); also see OSI model. This CAL-based communication profile for industrial systems is defined in CiA Draft Standard 301.

At this point, it should be mentioned that other profile definitions (e.g. DeviceNet, SDS, CANKingdom, etc.) are available on the market in addition to the CANopen profile.

The profile definitions for various devices have been drawn up according to the CANopen definition. The Device Profile for Encoders in CiA Draft Proposal 406 has been adopted for encoders.

The TWK encoder has been developed on the basis of the existing standard.

In each case, Class 1 and Class 2 functionality have been implemented. The latter contains the programmable version for encoders.

The profile is described in detail in the operating instructions. The prerequisite, however, is exact knowledge of the CAL and the CANopen profile.

2. CANopen Communication Profile

The different communication objects and communication services, and the different start and stop conditions for message transmission within a CANopen network are described in the DS 301, CANopen, Communication Profile.

CANopen supports the transmission of synchronous and asynchronous messages. As a result of synchronous message transmission, network-wide data capture and processing with universal reference is possible.Defined communication objects for this purpose are (SYNC-message, Time Stamp Message).The transmission of the synchronisation object is carried out via the SYNC-master, which controls network-wide data capture, processing and transmission.

Asynchronous messages are result-dependent and may be transmitted at any desired point in time. Differentiation is made between four functionally-dependent message groups:

- Administration Messages (Layer Management, Network Management, Identifier Distribution Management)
- Service Data Messages
- Process Data Messages
- Pre-defined Messages (Synchronisation, Time Stamp, Emergency)

Administration Messages refer to those messages which are exchanged on use of CAL services for network configuration, network administration and identifier allocation. The services and protocols LMT, NMT, DBT and CMS, and are contained in the CAL standard.The Service Data Messages are exchanged in the event of direct write or read access to object directory entries. The objects which are to be exchanged are designated as the Service Data Object. The main application of this type of message is in device configuration.The real-time transmission of current process data is realised with the Process Data Messages. The objects which are transferred are designated as the Process Data Object.

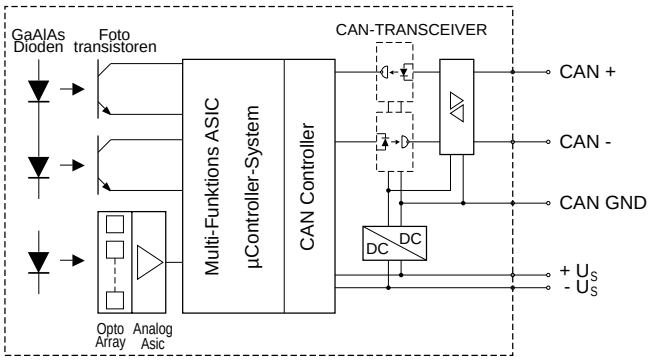
The fundamental characteristics of PDO and SDO are compared in the following table.

Process Data Object PDO	Service Data Object SDO
Is used for real-time data exchange.	No real-time data exchange. Enables access to any desired object directory entry. The SDO makes a peer-to-peer communication channel available.
Is typically a high-priority message:	Is generally a low-priority message.
Can be transmitted both synchronously and asynchronously:	Transmission is typically asynchronous and is confirmed with a service.
Both cyclical and acyclical message transmission are possible.	Acyclical transmission. Transmission is initiated by the client.
The data content of the PDO is defined in the object directory, and can be configured via an SDO.	Use of the data field is determined by the CMS Multiplexed Domain Protocol.
Data field format is specified.	Access to an object directory entry of a device is carried out by specifying index and subindex (direct addressing).

3. CANopen scope of the TWK Encoder

- According to Device Profile DS 406
- NMT-slave
- One SDO per communication direction for accessing
- Two PDOs per communication direction the object directory
- PDO modes: synchronous / asynchronous
- Identifier setting via Node ID (default)
- PDO-Identifier can be adjusted via SDO
- COB-ID distribution: default
- SYNC message
- EMERGENCY message
- Node guarding
- Simple boot-up according to DS 301
- Transmission Types can be adjusted for all PDOs
- CAN-LED support for bus state
- DIP-switch support for setting the node address (6bits) and the Baudrate (2 bits) for the connecting cap version
- Node ID and Baudrate setting across manufacturer-specific object range by means of SDO
- Encoder programming parameters: Code sequence, single turn resolution, total measuring range in measuring units, preset value, lower and upper work area

Schematic circuit diagram

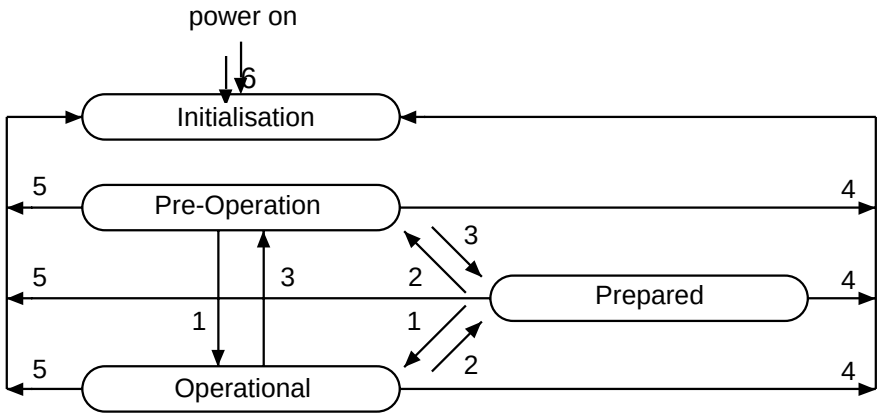


4. Network Management

4.1 Minimum-Capability-Device Boot up

The encoders are designed as Minimum Capability Devices according to CANopen (see CiA-DS 302), i.e. they behave according to the following state diagram:

For switching between the different states, the following CANopen services are required or are independently carried out by the encoder:



(1) Start Remote Node (start the CANopen node)
(2) Stop Remote Node (stop the CANopen node)
(3) Enter Pre-Operational (switch to pre-operational)
(4) Reset Node (reset the entire CANopen node)
(5) Reset Communication (reset CANopen node communication)
(6) Initialisation finished (initialisation terminated, automatic change of state to pre-operational)

In this case, it must be noted that the state Prepared must not be used on Boot-up.

The individual states can be described as follows:

Initialisation: Initial state following application of the supply voltage. After running through the initialisation routines, the encoder automatically switches to Pre-Operational state.

Pre-Operational: The node can be addressed via SDO messages under the standard identifier. The encoder and communication parameters can be programmed.

Operational: The encoder is active. The position value of the encoder is output via the PDOs.

Prepared: In this state, the node is no longer active, i.e. both SDO and PDO communication are impossible. The encoder can be set to the Operational or Pre-Operational state via the corresponding NMT commands.

The most simple method of starting an encoder as a CANopen subscriber is to execute the Start Remote Node service. In doing so, the encoder is brought into its Operational state.

4.2 Description of the NMT Commands

The NMT object is structured as follows:

	Byte0	Byte1
COB-ID = 0	Command byte cs	Node ID*

* If a 0 is entered for the node ID, the service is transmitted to all subscribers.

Command byte	**Description**
cs = 1	Start Remote Node: Switch to Operational
cs = 2	Stop Remote Node: Switch to Prepared
cs = 128	Enter Pre-Operational State: Switch to Pre-operational
cs = 129	Reset Node[1)]
cs = 130	Reset Communication[2)]

[1)] The parameters of the entire object directory are set to power-on values

[2)] The parameters of the object directory, communication profile section, are set to power-on values.

Example of switching nodes with node ID 1 to active:

Master -> Slave

ID	**DLC**	**Data**	
		Byte 0	Byte 1
NMTZeroMsg (=0)	2	1	1
(Software: CANalyzer)			

Slave -> Master

No confirmation or acknowledgement!

5. Node Guarding

The guarding of subscribers in the bus, especially in the case of event-controlled processes, is a sensible feature. A differentiation is made between lifeguarding and node guarding. In the case of lifeguarding, the NMT-master is also guarded by the NMT-slaves, in addition to the guarding of the slaves by the NMT-master.
Node Guarding alone is implemented in the encoder. This means that the NMT-master guards the NMT-slave via corresponding services.

On activation of node guarding, the network master transmits an RTR-telegram with the COB-ID for node guarding (700H + node ID) to the node at regular intervals (guard time).
The slave-NMT responds under COB-ID with the status in Byte 0.

Byte 0

Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0

Bit 7: Toggle-Bit

If this bit does not change between two successive responses, an error has occurred. The value at the start (1st response) of guarding is „0“.

(The toggle bit is switched in the case of successive data frames, in order to guarantee that the application, and not only the CAN-Controller, responds to guarding)

Bit 0-6: Status

4: Prepared

5: Operational

127: Pre-Operational

The guard time (object 100CH), multiplied with the life time factor (object 100DH), reveals the life time period for the node guarding protocol.

If the subscriber does not respond within the life time period, corresponding application-specific precautions must be implemented. An example of this is, e.g., the implementation of a change of state from Operational to Prepared for the defective node.

Example:

1. Guard time (100CH) and life time factor (100DH) = 0
   Encoder in state: Operational

   NMT-guarding (e.g. Id:702 DLC: R1 Data: 0 see CANalyzer)
   1st response: Data: 05H
   NMT-guarding (e.g. Id:702 DLC: R1 Data: 0 see CANalyzer)
   2nd response: Data: 85H
   NMT-guarding (e.g. Id:702 DLC: R1 Data: 0 see CANalyzer)
   3rd response: Data: 05H

2. Guard time (100CH) = 100H and life time factor (100DH) = 1H
   Encoder in state: Operational

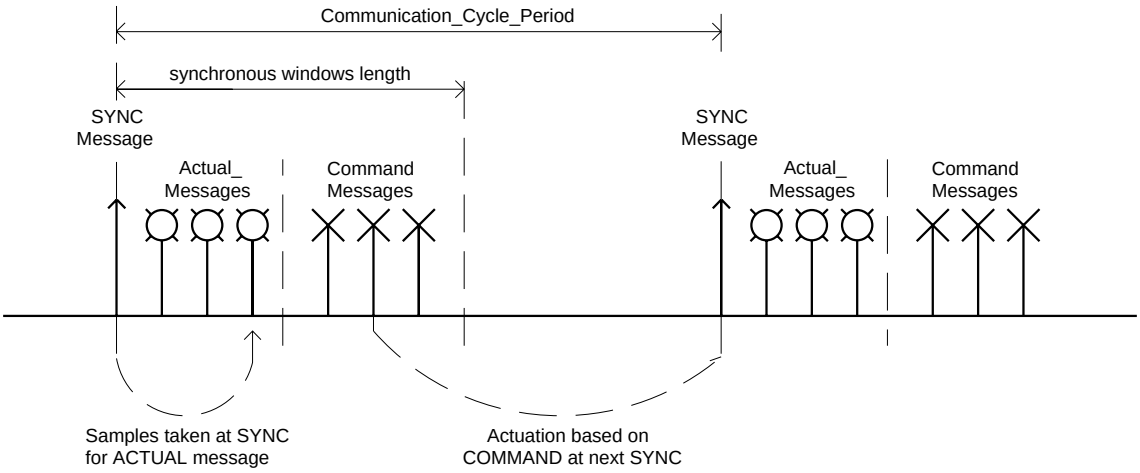
   1st NMT-guarding (e.g. Id:702 DLC: R1 Data: 0 see CANalyzer) within the life time
   1st response: Data: 05H
   2nd NMT-guarding (e.g. Id:702 DLC: R1 Data: 0 see CANalyzer) within the life time
   2nd response: Data: 85H
   3rd NMT-guarding (e.g. Id:702 DLC: R1 Data: 0 see CANalyzer) within the life time
   3rd response: Data: 05H
   4th NMT-guarding (e.g. Id:702 DLC: R1 Data: 0 see CANalyzer) **not** within the life time
   4th response: Data: FFH
   Encoder independently switches into Pre-Operational state!

Changes in state on failure of guarding:

Prepared	->	Pre-Operational
Operational	->	Pre-Operational
Pre-Operational	->	Pre-Operational

6. Bus Synchronisation

The CANopen communication profile enables the synchronisation of subscribers in the bus. In this case, the SYNC-message (also see object 1005H, default value = 80H) serves to provide synchronisation between the various bus subscribers. In the case of real-time applications, certain procedures must be time-synchronised and must run in certain pre-specified time windows.
The time between two SYNC-telegrams is described as the communication cycle. Within a communication cycle, differentiation is made between the report window (actual values: e.g. actual encoder position values, I/O module) and command window (nominal values or initial values: e.g. commands to drives, actuation command for actuators) time windows. The nominal values and commands become actual in the case of the next SYNC-telegram. Following the SYNC-window, low-priority messages are able to access the bus.



7. Emergency Message-supported Error Codes

In the event of an error, an emergency telegram is output via the COB-ID emergency message (see object: 1014H - standard: 80H + node-ID). This contains the emergency error code (byte 0,1), see below and the error register (byte 2). The error register is described under object 1001H.

Emergency Error code (hex)	**Designation**	**Comment**
0000	Error reset or no error	Error reset or no error present
5000	Device hardware	MSA and/or expander error
5100	Hardware memory error	ROM code error
8100	Communication	Communication error (Master - Encoder communication e.g. interrupted)
FFFF	Device-specific	Manufacturer-specific: see object 6503 Alarms

Object 6503 Alarms
Meaning in the case of emergency message FFFFH

Bit	**Designation/value**	**Supported**
Bit 11-0	0	no
Bit 12	EEPROM error	yes
Bit 13	CRC error EEPROM	yes
Bit 14	XRAM error	yes
Bit 15	0	no

Object 6504 Supported Alarms (see above)

8. Installation Instructions

8.1 Node ID, Baudrate, Terminal Resistor

Connecting cap version:

DIP-switch:

DIP-switch	1	2	3	4	5	6	7	8	9	10
ON		x							x	x
OFF	x		x	x	x	x	x	x		

Configuration:

		Range of values		Example
Node ID:	Switch 1 - 6	1-64		**Value + 1 = Node ID = 3**
Baudrate:	Switch 7-8	0x0	20 kBaud	**20 kBaud**
		0x1	125 kBaud	
		0x2	500 kBaud	
		0x3	1MBaud	
Terminal Resistor:	Switch 9-10	On	On	**Terminal resistor switched on**
				(see example)
		Off	Off	Terminal resistor not switched on

Plug version:

Default values (on delivery)	Baudrate:	20 kBaud
	Default address:	1
Value range:	Baudrate	0x0 20 kBaud
		0x1 125 kBaud
		0x2 500 kBaud
		0x3 1MBaud
	Node address:	1 - 64

Procedure on changing parameters:

1. Edit parameter (object: 2000H Node address
   2001H Baudrate)
2. Store parameter (object 1010H “SAVE”)
3. Reset node
4. New parameter values are actual

8.2 LED Description:

UB	Data	COM	Err	Comment
green	green	green	red	
ON				Operating voltage present
	flash			Process data communication
		ON		CAN communication OK.
		flash		CAN communication Warning state
	OFF			CAN communication BUS OFF
			ON	Error which triggers the Emergency Protocol
				or SDO Abort Transfer Protocol

9. The Object Directory

9.1 Subdivision of the Object Directory

Index (hex)	Object
0000	unused
0001-001F	static data types
0020-003F	complex data types
0040-005F	manufacturer-specific data types
0060-0FFF	reserved
1000-1FFF	communication profile
2000-5FFF	manufacturer-specific profile
6000-9FFF	standardised device profile
A000-FFFF	reserved

9.2 Services for Writing and Reading the Object Directory

Access to the object directory of the encoder is carried out via SDO messages.
The device and communication-specific parameters of the encoder are described in the object directory

The structure of the SDO message (expedited SDO, may 4 bytes) is as follows:

COB-ID	Command	Index		Subindex	Service data			
11 bit	Byte 0	Byte 1 (LSB)	Byte 2 (MSB)	Byte 3	Byte 4 (LSB)	Byte 5	Byte 6	Byte 7 (MSB)

The following identifiers are available as standard for the SDO services:

SDO (tx) (slave -> master): 580H (1408) + Node ID
SDO (rx) (master -> slave): 600H (1536) + Node ID
(Note: The SDO identifiers cannot be changed)

9.3 COB-ID Structure

The COB-ID - COB identifier serves to clearly identify a CAN message. The data are transmitted throughout the network via communication objects (COB).
The identifier is 11 bits wide, and is comprised of a 4 bit function code and a 7 bit node ID. The priority of a message decreases with an ascending COB identifier.

COB-ID structure, see Predefined Master/Slave Connection Set according to CANopen:

Bit No.:	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Function code				**Node ID**						
Assignment	x	x	x	x	0	x	x	x	x	x	x
x - 0 or 1											

9.4 Message Objects and Function Codes

On use of the Predefined Master/Slave Connection Set (see DS 301 CANopen), the message identifiers are defined as follows:

Object	Function code	Resulting COB-ID		Communication parameter parameter from index	CMS priority
	(binary)	Hexadecimal	Decimal	(hexadecimal)	group
NMT	0000	0	0	-	0
SYNC	0001	80	128	1005	0
Emergency	0001	81-FF	129-255	-	0,1
PDO 1(tx)	0011	181-1FF	385-511	1800	1,2
PDO 2(tx)	0101	281-2FF	641-767	1802	2,3
SDO (tx)	1011	581-5FF	1409-1535	-	6
SDO (rx)	1100	601-67F	1537-1663	-	6,7
Nodeguard	1110	701-77F	1793-1919	100E	-

9.5 The Command Byte Describes the Type of SDO Message (see examples for SDOs)

Command in hex	Type	Function
22	SDO (rx) Initiate download request (data length max. 4 bytes)	Transmit parameter to encoder
23	SDO (rx) Initiate download request (data length = 4 bytes)	Transmit parameter to encoder
2B	SDO (rx) Initiate download request (data length = 2 bytes)	Transmit parameter to encoder
2F	SDO (rx) Initiate download request (data length = 1 byte)	Transmit parameter to encoder
60	SDO (tx) Initiate download response	Confirmation of take-over to master
40	SDO (rx) Initiate upload request	Request parameter from encoder
43	SDO (rx) Initiate upload response	Parameter to master with data length = 4 bytes (Unsigned 32)
4B	SDO (rx) Initiate upload response	Parameter to master with data length = 2 bytes (Unsigned 16)
4F	SDO (rx) Initiate upload response	Parameter to master with data length = 1 byte (Unsigned 8)
80	SDO (rx) Abort domain transfer	Encoder reports error code to master

9.6 Examples of Service Data Objects

Reading the object directory

Master -> Slave

COB-ID+ NodeID	**Byte0**	**Byte1**	**Byte2**	**Byte3**	**Byte4**	**Byte5**	**Byte6**	**Byte7**
		Index		Subindex	reserved			
600H + Node-ID	40H	LSB	MSB					

Slave -> Master

COB-ID+ NodeID	**Byte0**	**Byte1**	**Byte2**	**Byte3**	**Byte4**	**Byte5**	**Byte6**	**Byte7**
		Index		Subindex	reserved			
580H + Node-ID	42H	LSB	MSB					

Example: Reading the position value (object 6004H)

COB-ID+ NodeID	**Byte0**	**Byte1**	**Byte2**	**Byte3**	**Byte4**	**Byte5**	**Byte6**	**Byte7**
601	40	04	60	00	00	00	00	00
581	43	04	60	00	34	12		

Writing the object directory

Master -> Slave

COB-ID + NodeID	**Byte0**	**Byte1**	**Byte2**	**Byte3**	**Byte4**	**Byte5**	**Byte6**	**Byte7**
		Index		Subindex	reserved			
600H + Node-ID	22H	LSB	MSB					

Slave -> Master

COB-ID + NodeID	**Byte0**	**Byte1**	**Byte2**	**Byte3**	**Byte4**	**Byte5**	**Byte6**	**Byte7**
		Index		Subindex	reserved			
580H + Node-ID	60H	LSB	MSB					

Example: Writing the preset value 1000H (object 6003H)

COB-ID+ NodeID	**Byte0**	**Byte1**	**Byte2**	**Byte3**	**Byte4**	**Byte5**	**Byte6**	**Byte7**
601	22	03	60	00	00	10	00	00
581	60	03	60	00	00	10		

10. Communication Profile

Communication Profile Overview / Communication-Profile

Index (hex)	Object Code	ObjectName	Type	Attr.	C1	C2
1000	VAR	Device type	unsigned32	const	m	
1001	VAR	Error register	unsigned8	ro	m	
1002	VAR	Manufacturer status register	unsigned32	ro	o	o
1003	ARRAY	Pre-defined error field	unsigned32	ro	o	o
1004	ARRAY	Number of PDOs supported	unsigned32	ro	o	o
1005	VAR	COB-ID SYNC-message	unsigned32	rw	o	o
1008	VAR	Device name	Vis-String	const	o	o
1009	VAR	Hardware version	Vis-String	const	o	o
100A	VAR	Software version	Vis-String	const	o	o
100B	VAR	Node-ID	unsigned32	ro	o	o
100C	VAR	Guard time	unsigned32	rw	o	o
100D	VAR	Life time factor	unsigned32	rw	o	o
100E	VAR	COB-ID guarding protocol	unsigned32	rw	o	o
100F	VAR	Number of SDO´s supported	unsigned32	ro	o	o
1010	VAR	Store parameters	unsigned32	rw	m	m
1011	VAR	Restore Default Parameters	unsigned32	rw	m	m
1014	VAR	COB-ID Emergency message	unsigned32	rw	o	o
1 st Transmit PDO (asynchronous transmission)						
1800		1st transmit PDO Parameter	PDO CommPar	rw	m	m
1A00		TxPDO 1 Mapping	PDO Mapping	ro	o	o
2 nd Transmit PDO (synchronous transmission)						
1802		2nd transmit PDO Parameter	PDO CommPar	rw	m	m
1A02		TxPDO 2 Mapping	PDO Mapping	ro	o	o

10.1 Object 1000H: Device Type

Object 1000H	**Device Type** The object 1000H contains the device profile number and the encoder type according to /1/	
Object description	**Index**	1000H
	Name	Device Type
	Object Code	VAR
	Data Type	unsigned32
Value description	**Access**	ro (const)
	Value Range	unsigned32
	Default Value	no

Structure of the parameters

Device Type			
Device Profile Number		Encoder Type	
Byte 0	Byte 1	Byte 2	Byte 3
96	1	2	0
		Multiturn absolute rotary encoder	

Description: Encoder: 0x00020196

10.2 Object 1001H: Error Register

Object 1001H

Error Register

The error register displays encoder device errors.

Object description

Index	1001H
Name	Error Register
Object Code	VAR
Data Type	unsigned8

Value description

Access	ro
Value Range	unsigned8
Default Value	0x0 (no Error)

Description

Bit	mandatory/ optional	Meaning	supported
0	m	generic error	yes
1	o	current	no
2	o	voltage	no
3	o	temperature	no
4	o	communication error (overrun, error state,...)	yes
5	o	device profil specific	no
6	o	reserved	no
7	o	manufacturer specific	yes

Default value: 00H (no error)

10.3 Object 1002H: Manufacturer Status Register

Object 1002H	**Manufacturer Status Register** This object is the manufacturer status register.	
Object description	**Index**	1002H
	Name	Manufacturer Status Register
	Object Code	VAR
	Data Type	unsigned32
Value description	**Access**	ro
	Value Range	unsigned32
	Default Value	00H
Description	Not further supported at present.	

10.4 Object 1003H: Predefined Error Field

Object 1003H

Predefined Error Field
The device errors are entered into object 1003H. - Subindex 00H Number of Errors stored (max. 1). - Subindex 01H Each new error is stored under subindex 01. The error display is deleted by writing the value zero into subindex 00H (only when the error state has been eradicated).

Object description

Index	1003H
Name	Predefined Error Field
Object Code	Array
Data Type	unsigned32

Value description

Sub-Index	00H
Description	Number of Errors
Access	rw
Value Range	unsigned8

Sub-Index	01H
Description	Standard Error Field
Access	ro
Value Range	unsigned32
Default Value	00H

Example:

e.g. **EEPROM error**

Error register	1001H	0x81
Predefined Error Field	1003,00H	0x1
	1003,01H	0x81FFFF
Alarms	6503H	0x1000

e.g. **no error**

Error register	1001H	0x0
Predefined Error Field	1003,00H	0x0
	1003,01H	0x0
Alarms	6503H	0x0

10.5 Object 1004H: Number of PDOs Supported

Object 1004H	**Number of PDOs supported** Index 1004H contains the maximum number of PDOs which are supported by the encoder. Differentiation is made between synchronous/asynchronous and transmit/receive PDOs in this case.	
Object description	**Index**	1004H
	Name	Number of PDOs supported
	Object Code	Array
	Data Type	unsigned32
Value description	**Sub-Index**	00H
	Description	Number of PDOs supported
	Access	ro
	Value Range	unsigned32
	Default Value	0x2H
	Sub-Index	01H / 02H
	Description	01H: Number of synchronous PDOs 02H:Number of asynchronous PDOs
	Access	ro
	Value Range	unsigned32
	Default Value	0x1/ 0x1

Description

Subindex	**MSB**	**LSB**
00H	Number of receive PDOs	Number of transmit PDOs
01H	Number of synchr. receive PDOs	Number of synchr. transmit PDOs
02H	Number of asynchr. receive PDOs	Number of asynchr. transmit PDOs

Subindex	**Value**	**Description**
00	02H	2 transmit PDOs
01	01H	1 synchronous transmit PDO
02	01H	1 asynchronous transmit PDO

10.6 Object 1005H: COB-ID SYNC Message

Object 1005H	**COB-ID SYNC Message**	
	The object 1005H defines the COB-ID for the SYNC message (synchronisation message). In addition, it is defined whether the subscriber generates SYNC messages or merely reacts to SYNCs.	
Object description	**Index**	1005H
	Name	COB-ID SYNC Message
	Object Code	VAR
	Data Type	unsigned32
Value description	**Access**	rw
	Value Range	unsigned32
	Default Value	80H

Parameter structure

Bit	Value	Meaning
31(MSB)	0	Device does not consume SYNC message
	1	Device consumes SYNC message
30	0	Device does not generate SYNC message
	1	Device generates SYNC message
29	0	11-Bit ID (CAN 2.0A)
	1	29-Bit ID (CAN 2.0B)
28-11	0	if bit 29 = 0
	x	if bit 29 = 1; Bits 28-11 of 29-bit-SYNC-COB-ID
10-0(LSB)	x	bits 10-0 of SYNC-COB-ID
		(**Standard-ID = 80H**)

Example Encoder 0x80000080
(Note: On changing the COB-ID, the current value only becomes valid following storage (see object 1010H) and subsequent encoder resetting).

10.7 Object 1006H: Communication Cycle Period

Object 1006H	**Communication Cycle Period** Object 1006H defines the communication cycle period in ms. If the value is 0, this object is not used.	
Object description	**Index**	1006H
	Name	**Communication Cycle Period**
	Object Code	VAR
	Data Type	unsigned32
Value description	**Access**	rw
	Value Range	unsigned32
	Default Value	0H

10.8 Object 1007H: Synchronous Window Length

Object 1007H	**Synchronous Window Length** Object 1007H defines the synchronous window length parameter in ms. If the value is 0, this object is not used.	
Object description	**Index**	1007H
	Name	**Synchronous Window Length**
	Object Code	VAR
	Data Type	unsigned32
Value description	**Access**	rw
	Value Range	unsigned32
	Default Value	0H

10.9 Object 1008H: Device Name

Object 1008H	**Device Name**	
	Object 1008H contains the manufacturer-specific device name.˝	
Object description	**Index**	1008H
	Name	**Manufacturer Device Name**
	Object Code	VAR
	Data Type	Visible String
Value description	**Access**	ro
	Default Value	Encoder CRN

10.10 Object 1009H: Manufacturer Hardware Version

Object 1009H	**Manufacturer Hardware Version**	
	Object 1009H contains the hardware version number.	
Object description	**Index**	1009H
	Name	**Manufacturer Hardware Version**
	Object Code	VAR
	Data Type	Visible String
Value description	**Access**	ro
	Default Value	e.g. 01.00

10.11 Object 100AH: Manufacturer Software Version

Object 100AH	**Manufacturer Software Version**	
	Object 100AH contains the software version number.	
Object description	**Index**	100AH
	Name	**Manufacturer Software Version**
	Object Code	VAR
	Data Type	Visible String
Value description	**Access**	ro
	Default Value	e.g. 01.40

10.12 Object 100BH: Node ID

Object 100BH	**Node ID**	
	Object 100BH contains the node ID. Access to the node ID entry is of the “read only” type, and cannot be changed via SDO. This entry provides the value which is stored in EEPROM.	
Object description	**Index**	100BH
	Name	Node ID
	Object Code	VAR
	Data Type	unsigned32
Value description	**Access**	ro
	Value Range	e.g.: 0x5

Parameter structure

MSB			LSB
reserved	reserved	reserved	Node-ID

Description

The node ID may assume values from 1 to 64.
In the case of the connecting cap version, the node ID value is calculated as follows:

Node ID = DIP-switch value + 1

In the case of the plug version, the node ID is changed via the object 2000H.

10.13 Object 100CH: Guard Time

Object 100CH	**Guard Time**	
	Contains the guard time in ms. The life time factor (object 100DH), multiplied by the guard time, reveals the life time for node guarding.	
Object description	**Index**	100CH
	Name	Guard Time
	Object Code	VAR
	Data Type	unsigned16
Value description	**Access**	rw
	Value Range	unsigned16
	Default Value	0H

10.14 Object 100DH: Life Time Factor

Object 100DH	**Life time factor:**	
	The life time for node guarding is revealed when the life time factor is multiplied by the guard time. Example: Guard time: 100 ms Life time factor: 5 The guard time of 100 ms allows the nodes in the network to expect a guard frame from the master every 100 ms. The frequency with which a guard frame from the master may be omitted without an error state's being recognised is set for each module via the life time factor. In this example, the relevant time would therefore be 500 ms. The omitted message would only be correspondingly evaluated after 500 ms.	
Object description	**Index**	100DH
	Name	Life Time Factor
	Object Code	VAR
	Data Type	unsigned8
Value description	**Access**	rw
	Value Range	unsigned8
	Default Value	0H

10.15 Object 100EH: Node Guarding Identifier

Object 100EH	**Node Guarding Identifier** Object 100EH contains the identifier for node guarding.	
Object description	**Index**	100EH
	Name	Node Guarding Identifier
	Object Code	VAR
	Data Type	unsigned32
Value description	**Access**	rw
	Value Range	unsigned32
	Default Value	700H + Node-ID / 1792 + Node-ID

(Note: On changing the COB-ID, the current value only becomes valid following storage (see object 1010H) and subsequent encoder resetting).

10.16 Object 100FH: Number of SDO´s supported

Object 100FH	**Number of SDOs Supported** Object 100FH contains the number of SDOs which are supported.	
Object description	**Index**	100FH
	Name	**Number of SDOs Supported**
	Object Code	VAR
	Data Type	unsigned32
Value description	**Access**	rw
	Value Range	unsigned32
	Default Value	1H

Parameter structure
MSB Number of client SDOs
LSB Number of server SDOs

Example
Encoder with node ID 1
SDO (tx) (slave -> master): 580H (1409) + node ID: 581H
SDO (rx) (master -> slave): 600H (1537) + node ID: 601H
(Note: The SDO identifiers cannot be changed)

10.17 Object 1010H: Store Parameters

Object 1010H

Store Parameters

Parameter changes can be stored via object 1010H.

Store Parameters:

Subindex 01H: In order to store the parameters, the word save must be entered.

Signature	MSB			LSB
ASCII	e	v	a	s
hex	65H	76H	61H	73H

Example: 0x65766173

The values only become valid after resetting!

Object description

Index	1010H
Name	Store Parameters
Object Code	Array
Data Type	unsigned32

Value description

Sub-Index	00H
Description	Largest supported Sub-Index
Access	ro
Value Range	unsigned8
Default Value	01H

Sub-Index	01H
Description	Save all Parameters
Access	rw
Value Range	unsigned32
Default Value	save' on storage

10.18 Object 1011H: Restore Default Parameters

Object 1011H

Restore Default Parameters
The default parameters can be restored via object 1011H. **Load parameters** Subindex 01H: In order to restore the parameters, the word **load** must be entered. Signature MSB LSB ASCII d a o l hex 64H 61H 6FH 6CH Example: 0x64616F6C **The values only become valid after resetting!**

Object description

Index	1011H
Name	Restore default Parameters
Object Code	Array
Data Type	unsigned32

Value description

Sub-Index	00H
Description	Largest supported Sub-Index
Access	ro
Value Range	unsigned8
Default Value	01H

Sub-Index	01H
Description	Restore all default Parameters
Access	rw
Value Range	unsigned32
Default Value	load' on restoring

10.19 Object 1014H: COB-ID Emergency Message

Object 1014H	**COB-ID Emergency Message** Object 1014H contains the COB-ID for the emergency message.	
Object description	**Index**	1014H
	Name	**COB-ID Emergency Message**
	Object Code	VAR
	Data Type	unsigned32
Value description	**Access**	rw
	Value Range	unsigned32
	Default Value	80H + Node-ID

(Note: On changing the COB-ID, the current value only becomes valid following storage (see object 1010H) and subsequent encoder resetting).

10.20 Overview of Transmission Types

Transmission Type					
Code	Transmission Type				
	cyclical	acyclical	synchronous	asynchronous	RTR only
0		x	x		
1-240	x		x		
241-251	reserved				
252			x		x
253				x	x
254				x	
255				x	

Meaning:

0: After SYNC but only in the event of a change in value since the last SYNC
1 - 240: Transmit value after 1st and 240th SYNC message
252: SYNC leads to internal value storage, value must be collected via RTR
253: Value is updated and transmitted after RTR
254: Value is updated and transmitted after a change in value (if device timer (cyclical timer) = 0) or following expiry of the cycle time (device timer > 0)

10.21 Object 1800H: 1st Transmit PDO (Asynchronous Transmission)

Object 1800H

1st Transmit PDO (Asynchronous Transmission)

Object 1800H contains the communication parameters for the transmit PDO. This PDO transmits the encoder position data asynchronously.The cyclical timer is stored in object 6200H.

Object description

Index	1800H
Name	1st Transmit PDO (Asynchronous Transmission)
Object Code	RECORD
Data Type	PDOCommPar

Value description

Sub-Index	00H
Description	Number of supported entries, default:3
Access	ro
Value Range	unsigned8
Default Value	2 - 4

Sub-Index	01H
Description	COB-ID used by PDO and release
Access	rw
Value Range	unsigned32
Default Value	Index 1800H: 180H + Node-ID

Sub-Index	02H
Description	Transmission Type default: 254
Access	rw
Value Range	unsigned8
Default Value	0xFE

Sub-Index	03H
Description	Inhibit time, default: 0
Access	rw
Value Range	unsigned16
Default Value	0x0

Subindex 01:	**Bit 0 - 10**	**11 bit ID: 180H + node ID**
	Bit 11 - 29	0 (reserved 29-bit ID)
	Bit 30	0 RTR enabled (cannot be changed)
	Bit 31	0 PDO enabled (default)
		1 PDO disabled

Index	**Subindex**	**Designation**	**Default value**
1800H	0	Number of entries	3
	1	COB-ID PDO 1 and release	0x180 + node ID
	2	Transmission type	254
	3	Inhibit time*	0

* Minimum waiting period before this PDO may be retransmitted

10.22 Object 1A00H: 1st Transmit PDO (Asynchronous Transmission) Mapping

Object 1A00H

1st Transmit PDO (Asynchronous Transmission) Mapping

Object 1A00H contains the parameters for mapping the 1st TxPDO.In our case, this is the position value parameter.

Object description

Index	1A00H
Name	1st Transmit PDO (Asynchronous Transmission) Mapping
Object Code	RECORD
Data Type	PDOMapping

Value description

Sub-Index	00H
Description	Number of mapped objects, default:1
Access	ro
Value Range	unsigned8

Sub-Index	01H
Description	Position value
Access	ro
Value Range	unsigned32

Index	**Subindex**	**Designation**		**Default value**
1A00H	0	Number of entries		1
	1	Position value		60040020H
Parameter structure		MSB	Index (16 bit)	6004
			Subindex (8 bit)	00
		LSB	Object length (8 bit)	20 (see Table 10-4, DS 301)

10.23 Object 1802H: 2nd Transmit PDO (Synchronous Transmission)

Object 1802H	**2nd Transmit PDO (Synchronous Transmission)** On request, object 1802H cyclically transmits the position data of the encoder. The request is carried out via a remote frame and/or via a SYNC-telegram.	
Object description	**Index**	1802H
	Name	1st Transmit PDO (Asynchronous Transmission)
	Object Code	RECORD
	Data Type	PDOCommPar
Value description	See above (object 1800H)	

Index	**Subindex**	**Designation**	**Default value**
1802H	0	Number of entries	3
	1	COB-ID PDO 1 and release*	0x280 + node ID
	2	Transmission type	1
	3	Inhibit time	0
*Subindex 01:		Bit 0 - 10	11-bit ID: 280H + node ID
		Bit 11 - 29	0 (reserved 29-bit ID)
		Bit 30	0 RTR enabled (cannot be changed)
		Bit 31	0 PDO enabled (default) 1 PDO disabled

10.24 Object 1A02H: 2nd Transmit PDO (Asynchronous Transmission) Mapping

Object 1A02H

2nd Transmit PDO (Asynchronous Transmission) Mapping

Object 1A02H contains the parameters for mapping the 2nd TxPDO.In our case, this is the position value parameter.

Object description

Index	1A02H
Name	2nd Transmit PDO (Asynchronous Transmission) Mapping
Object Code	RECORD
Data Type	PDOMapping

Value description

Sub-Index	00H
Description	Number of mapped objects, default:1
Access	ro
Value Range	unsigned8

Sub-Index	01H
Description	Position value
Access	ro
Value Range	unsigned32

Index	Subindex	Designation	Default value
1A02H	0	Number of mapped objects	1
	1	Position value	60040020h

(see above)

11. Encoder-Profile

11.1 Encoder Parameters

Encoder parameter overview

Index (hex)	Object code	Object name	Type	Attr.	C1	C2
6000	VAR	Operating parameters	unsigned16	rw	m/o	m/o
6001	VAR	Single turn resolution	unsigned32	rw		m
6002	VAR	Total measuring range in measuring units	unsigned32	rw		m
6003	VAR	Preset value	unsigned32	rw		m
6004	VAR	Position value	unsigned32	ro		m
6200	VAR	Cyclic Timer	unsigned16	rw		m
6400	ARRAY	Area State Register	unsigned8	ro		o
6401	ARRAY	Work Area low limit	integer32	rw		o
6402	ARRAY	Work Area high limit	integer32	rw		o

11.1.1 Object 6000H: Operating Parameters

Object 6000H

Operating Parameters

The encoder's operating parameters are set under object 6000H - operating parameters.
Operating parameters are:

- the code sequence CW/CCW
- encoder diagnostics (yes/no)
- release of the scaling function.

The **code sequence** CW (clockwise) or CCW (counter-clockwise) defines whether the position values increase (CW) or decrease (CCW) on rotation of the shaft in a clockwise direction (when looking towards the shaft).
Encoder diagnostics are not currently supported.
The **scaling function** is released via bit 2 = "1". The single turn resolution and total measuring range in measuring units parameters can be changed in an application-specific manner.

Object description

Index	6000H
Name	Operating Parameters
Object Code	VAR
Data Type	unsigned16
Access	rw

Parameter structure

Bit	Function	Bit = 0	Bit = 1	C1	C2	support.
0	Code Sequence	CW	CCW	m		yes
1	Commissioning Diagnostic Control	Disa.	Enab.	o		no
2	Scaling function control	Disa.	Enab.		m	yes
3-15	Reserved for further use					

Value	Meaning
00H	Code sequence CW Commissioning diagnostic control disable Scaling function control disable
01H	Code sequence CCW Commissioning diagnostic control disable Scaling function control disable
02H	Not currently supported
03H	Not currently supported
04H	Code sequence CW Commissioning diagnostic control disable Scaling function control enable
05H	Code sequence CCW Commissioning diagnostic control disable Scaling function control enable
06H	Not currently supported
07H	Not currently supported

11.1.2 Object 6001H: Measuring Units per Revolution

Object 6001H	**Measuring Units per Revolution** The measuring units per revolution parameter specifies the number of measuring units in reference to one revolution	
Object description	**Index**	6001H
	Name	Measuring Units per Revolution
	Object Code	VAR
	Data Type	unsigned32
	Access	rw
Value description	**Access**	rw
	Value Range	unsigned32
	Default Value	1000H / 2000H

Parameter structure

Measuring units per revolution			
Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

11.1.3 Object 6002H: Total Measuring Range in Measuring Units

Object 6002H

Total Measuring Range in Measuring Units
Object 6002H contains the total measuring range in measuring units. Total measuring range in measuring units = measuring units per revolution x measuring range. Measuring units per revolution: measuring units per revolution. Measuring range: number of revolutions.

Object description

Index	6002H
Name	Total Measuring Range in Measuring Units
Object Code	VAR
Data Type	unsigned32

Value description

Access	rw
Value Range	unsigned32
Default Value	1.000.000H / 2.000.000H

Parameter structure

Total measuring range in measuring units

Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

Note: If the encoder is used in endless operation, the total measuring range in measuring units may only be specified in 2^n powers (n = 1, 2, ..., 12).

(Example: Total measuring range in measuring units = 2^n power

Measuring units per revolution: 1024
Total measuring range in measuring units: 8192
Measuring range: 8

0 ... 8191, 0 ... 8191, 0 ... 8191, etc.

Total measuring range in measuring units # 2^n power

Measuring units per revolution: 1024
Total measuring range in measuring units: 3600
Measuring range: 3.5156

0 ... 3599, 0 ... 3599, 0 ... 4563*
* Measuring value jump dependent on actual physical position of the encoder.

11.1.4 Object 6003H: Preset Value

Object 6003H

Preset Value
The preset value is used to compare the machine zero point with the encoder zero point. To achieve this, the encoder position value is set to the preset value. An offset value, which takes the preset value into consideration, is calculated internally. If the values for code sequence, measuring units per revolution and total measuring range in measuring units are changed, the preset value is deleted and preassigned with the value 0.

Object description

Index	6003H
Name	Preset Value
Object Code	VAR
Data Type	unsigned32

Value description

Access	rw
Default Value	0H

Parameter structure

Preset value

Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

11.1.5 Object 6004H: Position Value

Object 6004H

Position Value
Object 6004H is the position value of the encoder.

Object description

Index	6004H
Name	Position Value
Object Code	VAR
Data Type	unsigned32

Value description

Access	ro
PDO Mapping	yes
Value Range	unsigned32
Default Value	no

Parameter structure

Position Value

Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

11.1.6 Object 6200H: Cyclic Timer

Object 6200H

Cyclic Timer
Object 6200H contains the parameter “Cyclical Timer”. The cyclical timer defines the cycle time for all asynchronous PDOs (also see object 1800H: 1st Transmit PDO). The cyclical transmission of the position value of the encoder is carried out at a value > 0. The range of values for the cyclical timer is between 1 and 65535 ms. Example: 1 ms = 1H 256 ms = 100H

Object description

Index	6200H
Name	Cyclic Timer
Object Code	VAR
Data Type	unsigned16

Value description

Access	rw
Value Range	unsigned16
Default Value	0H

11.1.7 Object 6400H: Area State Register

Object 6400H

Area State Register
Object 6400H, the area state register, displays the position of the position value in the work area. In this case, a distinguishment is made between a lower and upper work area and outside of the work area.

Object description

Index	6400H
Name	Area State Register
Object Code	ARRAY
Data Type	unsigned8

Value description

Sub-Index	00H
Description	Number of available channels
Access	ro
Default Value	01H

Sub-Index	01H
Description	Work Area State Channel 1
Access	ro
Value Range	unsigned8
Default Value	no

Parameter structure

Work area state

Bit	7	reserved
	6	reserved
	5	reserved
	4	reserved
	3	reserved
	2	range underflow
	1	range overflow
	0	out of range (also see object 650A)

Value	**Meaning**
0	Position value in the work area
1	Position value outside of the work area
2	Position value greater than the work area
4	Position value less than the work area

11.1.8 Object 6401H: Work Area Low Limit

Object 6401H

Work Area Low Limit
Object 6401H, the “Work Area Low Limit” is the lower limit of the work area. On reaching the lower limit of the work area, bit 2 of object 6400H/subindex 01 is set.

Object description

Index	6401H
Name	Work Area Low Limit
Object Code	ARRAY
Data Type	Integer32

Value description

Sub-Index	00H
Description	Number of available channels
Access	ro
Default Value	01H

Sub-Index	01H
Description	Work Area Low Limit Channel 1
Access	rw
Value Range	Integer32
Default Value	0xFFFFF

Parameter structure

Work area low limit

Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

11.1.9 Object 6402H: Work Area High Limit

Object 6402H

Work Area High Limit
Object 6402H, the “Work Area High Limit” is the upper limit of the work area. On reaching the upper limit of the work area, bit 1 of object 6400H/subindex 01 is set.

Object description

Index	6402H
Name	Work Area High Limit
Object Code	ARRAY
Data Type	Integer32

Value description

Sub-Index	00H
Description	Number of available channels
Access	ro
Default Value	01H

Sub-Index	01H
Description	Work Area High Limit Channel 1
Access	rw
Value Range	Integer32
Default Value	1.F00.000H

Parameter structure

Work area high limit

Byte 0	Byte 1	Byte 2	Byte 3
2^{7} to 2^{0}	2^{15} to 2^{8}	2^{23} to 2^{16}	2^{31} to 2^{24}

11.2 Encoder Diagnostics

Index (hex)	Object Code	ObjectName	Type	Attr.	C1	C2
6500	VAR	Operating Status	unsigned16	ro	m	
6501	VAR	Single Turn Resolution	unsigned32	ro	m	
6502	VAR	Number of distinguishable Revolutions	unsigned16	ro	m	
6503	VAR	Alarms	unsigned16	ro		m
6504	VAR	Supported Alarms	unsigned16	ro		m
6505	VAR	Warnings	unsigned16	ro		m
6506	VAR	Supported Warnings	unsigned16	ro		m
6507	VAR	Profile and Software version	unsigned32	ro		m
6508	VAR	Operating Time	unsigned32	ro		m
6509	VAR	Offset Value	integer32	ro		m
650A	REC	Module Identification	integer32	ro		m
650B	VAR	Serial Number	unsigned32	ro		m

11.2.1 Object 6500H: Operating Status

Object 6500H

Operating Status

The encoders status of the operating parameters are set under the object 6500h.

Object description

Index	6500H
Name	Operating Status
Object Code	VAR
Data Type	unsigned16
Access	ro

Parameter structure

Bit	Function	Bit = 0	Bit =1	C1	C2	Supp.
0	Code Sequence	CW	CCW	m		yes
1	Commissioning Diagnostic Control	Not Supp.	Supp.	o		no
2	Scaling function control	Disa.	Enab.		m	yes
3 - 15	Reserved for further use					no

11.2.2 Object 6501H: Single Turn Resolution

Object 6501H

Single Turn Resolution
The parameter single turn resolution specifies the number of measuring units in reference to a revolution (360°).

Object description

Index	6501H
Name	Single Turn Resolution
Object Code	VAR
Data Type	unsigned32

Value description

Access	ro
Value Range	unsigned32
Default Value	0x1000 / 0x2000

Parameter structure

Single Turn resolution

Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

11.2.3 Object 6502H: Number of Distinguishable Revolutions

Object 6502H

Number of Distinguishable Revolutions
Object 6502H contains the measuring range. Total measuring range in measuring units = measuring units per revolution x measuring range. Measuring units per revolution: measuring units per revolution. Measuring range: number of revolutions.

Object description

Index	6502H
Name	Number of Distinguishable Revolutions
Object Code	VAR
Data Type	unsigned16

Value description

Access	ro
Value Range	unsigned16
Default Value	1000H

Parameter structure

Number of distinguishable revolutions

Byte 0	Byte 1	Byte 2	Byte 3
2^7 to 2^0	2^{15} to 2^8	2^{23} to 2^{16}	2^{31} to 2^{24}

11.2.4 Object 6503H: Alarms

Object 6503H	**Alarms**	
	An alarm message is output in the event of an error. This message is only extinguished when the error is no longer	
Object description	**Index**	6503H
	Name	Alarms
	Object Code	VAR
	Data Type	unsigned16
	Access	ro

Parameter structure

Bit	Function	Supported
0	Position error	no
1	Commissioning diagnostics	no
2 - 11	Reserved for further use	-
12	EEPROM error	yes
13	CRC error EEPROM	yes
14	XRAM error	yes
15	Manufacturer-specific (not currently assigned)	-

11.2.5 Object 6504H: Supported Alarms

Object 6504H	**Supported Alarms**	
	Object 6504H displays the alarm messages which are supported by the encoder.(Also see object 6503H).	
Object description	**Index**	6504H
	Name	Supported Alarms
	Object Code	VAR
	Data Type	unsigned16
	Access	ro

Parameter structure (see object 6503H)

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Data	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	
	7				0				0				0				**7000H**

11.2.6 Object 6505H: Warnings

Object 6505H	**Warnings**	
	If the tolerances of relevant, internal encoder parameters are exceeded, this is displayed via warnings.	
Object description	**Index**	6505H
	Name	Warnings
	Object Code	VAR
	Data Type	unsigned16
	Access	ro

Parameter structure

Bit	Function		Supported
0	Frequency exceeded		no
1	Light control reserve		no
2	CPU watchdog status		no
3	Operating time limit warning		no
4	Battery charge		no
5	Reference point	0 - reached 1 - not reached	yes
6 - 11	Reserved for further use		-
12 - 15	Manufacturer specific functions		no

11.2.7 Object 6506H: Supported warnings

Object 6506H	**Supported Warnings**	
	Object 6506H displays the warning messages which are supported by the encoder. (Also see object 6505H).	
Object description	**Index**	6506H
	Name	Supported Warnings
	Object Code	VAR
	Data Type	unsigned16
	Access	ro

Parameter structure (see object 6505H)

Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
Data	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
	0				0				2				0				**20H**

11.2.8 Object 6507H: Profile and Software Version

Object 6507H

Profile and Software Version
Object 6507H contains the profile and software version which is implemented in the encoder.

Object description

Index	6507H
Name	Profile and Software Version
Object Code	VAR
Data Type	unsigned32

Value description

Access	ro
Value Range	unsigned32
Default Value	1000200H

Parameter structure

Profile version		Software version	
Byte 0	Byte 1	Byte 2	Byte 3
2^{7} to 2^{0}	2^{15} to 2^{8}	2^{7} to 2^{0}	2^{15} to 2^{8}

11.2.9 Object 6508H: Operating Time

Object 6508H

Operating Time
Object 6508H contains the encoder's operating time counter. The operating time counter is not currently supported, for which reason the value is pre-assigned with FF FF FF FFH.

Object description

Index	6508H
Name	Operating Time
Object Code	VAR
Data Type	unsigned32

Value description

Access	ro
Value Range	unsigned32
Default Value	FFFF.FFFFH

11.2.10 Object 6509H: Offset Value

Object 6509H

Offset Value
Object 6509H contains the offset value parameter. The offset value is the position value which is calculated taking the preset value into consideration. (Not currently supported).

Object description

Index	6509H
Name	Offset Value
Object Code	VAR
Data Type	signed32

Value description

Access	ro
Value Range	signed32
Default Value	no

Parameter structure

Offset value

Byte 0	Byte 1	Byte 2	Byte 3
2^{7} to 2^{0}	2^{15} to 2^{8}	2^{23} to 2^{16}	2^{31} to 2^{24}

11.2.11 Object 650AH: Module Identification

Object 650AH

Module Identification
Object 650AH contains the module identification parameter. This parameter is subdivided as follows: Subindex 00H - Manufacturer Offset Value ((Manufacturer) offset value referring to the encoder's mechanical zero point) Subindex 01H - Manufacturer Minimum Position Value (Lower work area) Subindex 02H - Manufacturer Maximum Position Value (Upper work area)

Object description

Index	650AH
Name	Module Identification
Object Code	RECORD

Value description

Sub-Index	00H
Description	Manufacturer Offset Value
Access	ro
Value Range	signed32
Default Value	0H

Sub-Index	01H
Description	Manufacturer min. Position Value
Object Class	optional
Access	ro
Value Range	signed32
Default Value	FFFFFH

Sub-Index	02H
Description	Manufacturer max. Position Value
Object Class	optional
Access	ro
Value Range	signed32
Default Value	1.F00.000H

11.2.12 Object 650BH: Serial Number

Object 650BH	**Serial Number**	
	Object 650BH is the serial number. As this is not currently supported, it is preassigned with FF FF FF FFH.	
Object description	**Index**	650BH
	Name	Serial Number
	Object Code	VAR
	Data Type	unsigned32
Value description	**Object Class**	C2 Mandatory
	Access	ro
	Value Range	unsigned32
	Default Value	FFFF.FFFFH

11.3 Manufacturer-specific Profile Area

Index (hex)	Object Code	Object Name	Type	Attr.	C1	C2
2000	VAR	Node ID (Plug Version)	unsigned8	rw	m	
2001	VAR	Baudrate (Plug Version)	unsigned8	rw	m	

11.3.1 Object 2000H: Node ID (Plug Version)

Object 2000H

Node ID (Plug Version)

Changing the node ID for the plug version is carried out via object 2000H. The new node ID only becomes valid following storage (Save, see object 1010H) and subsequent encoder resetting. On delivery, the following values are set as defaults:
Node ID: 1
Baudrate: 20 kBaud
(Note:Changes to objects 2000H and 2001H have no effect in the case of the connecting cap version.The values set in the cover are primarily valid).

Object description

Index	2000H
Name	Node ID (Plug Version)
Object Code	VAR
Data Type	unsigned8

Value description

Access	rw
Value Range	unsigned8
Default Value	01H (Plug Version)

11.3.2 Object 2001H: Baudrate (Plug Version)

Object 2001H

Baudrate (Plug Version)

Changing the Baudrate for the plug version is carried out via object 2001H. The new Baudrate only becomes valid following storage (Save, see object 1010H) and subsequent encoder resetting.

4 Baudrates are supported:

00H	20 kBaud
01H	125 kBaud
02H	500 kBaud
03H	1 MBaud

(Further Baudrates on enquiry)

On delivery, the following values are set as defaults:

Node ID: 1

Baudrate: 20 kBaud

(Note:Changes to objects 2000H and 2001H have no effect in the case of the connecting cap version.The values set in the cover are primarily valid).

Object description

Index	2001H
Name	Baudrate (Plug Version)
Object Code	VAR
Data Type	unsigned8

Value description

Access	rw
Value Range	unsigned8
Default Value	00H = 20kBaud (Plug Version)

12. Default Values for Programming Parameters

1000H		Device Type	0x00020196
1001H		Error Register	0x0
1002H		Manufacturer Status Register	0x0
1003H		Predefined Error Field	
	00H	Number of Errors	0x1
	01H	Standard Errror Field	0x0
1004H		Number of PDO's supported	
	00H	Number of PDO's supported	0x2
	01H	Number of synchronous PDOs	0x1
	02H	Number of asynchronous PDOs	0x1
1005H		COB-ID Sync-Messages	0x80000080
1006H		Communication Cycle Period	0x0
1007H		Synchronous Window Length	0x0
1008H		Device Name	Encoder CRN
1009H		Manufacturer Hardware Version	01.00
100AH		Manufacturer Software Version	01.40

100BH		Node-ID	0x1
100CH		Guard Time	0x0
100DH		Life Time Factor	0x0
100EH		Node Guarding Identifier	700H + Node-ID
100FH		Number of SDOs supported	0x1
1010H		Store Parameters	
	00H	Largest supported Subindex	0x1
	01H	Save all Parameters	0x65766173 (store)
1011H		Restoe Default Parameters	
	00H	Largest supported Subindex	0x1
	01H	Restore all Default Parameters	0x64616F6C (restore)
1014H		COB-ID Emergency Message	80H + Node-ID
1800H		1st Transmit PDO	
	00H	Number of supported Entries	0x3
	01H	Cob-ID used by PDO1	180H + Node-ID
	02H	Transmisson Type	0xFE
	03H	Inhibit Time	0x0
1802H		2st Transmit PDO	
	00H	Number of supported Entries	0x3
	01H	Cob-ID used by PDO2	280H + Node-ID
	02H	Transmisson Type	0x1
	03H	Inhibit Time	0x0
1A00H		1st Transmit PDO Mapping	
	00H	Number of mapped Objects	0x1
	01H	Position Value	0x60040020
1A02H		2nd Transmit PDO Mapping	
	00H	Number of mapped Objects	0x1
	01H	Position Value	0x60040020
6000H		Operating Prameters	0x0
6001H		Measuring Units per Revolution	0x1000 /0x2000
6002H		Total Measuring Range in Measuring Units	0x1000000 /0x2000000
6003H		Preset Value	0x0
6004H		Position Value	0xXXXX
6200H		Cyclic Timer	0x0
6400H		Area State Register	
	00H	Number of available Channels	0x1
	01H	Work Area State Channel 1	0x0
6401H		Work Area Low Limit	
	00H	Number of available Channels	0x1
	01H	Work Area Low Limit Channel 1	0x FFFFF
6402		Work Area High Limit	
	00H	Work Area High Limit Channel 1	0x1
	01H	Work Area High Limit Channel 1	0x1F00000
6500H		Operating Status	0x0
6501H		Single Turn Resolution	0x1000/0x2000
6502H		Number of distinguishable Revolutions	0x1000
6503H		Alarms	0x0
6504H		Supported Alarms	0x7000
6505H		Warnings	0x0
6506H		Supported Warnings	0x20
6507H		Profil and Software Version	
6508H		Operating Time	0xFFFFFFFF
6509H		Offset Value	0xXXXXX
650AH		Module Identification	
	00H	Manufacturer Offset Value	0x0
	01H	Manufacturer min Position Value	0xFFFFF

	02H	Manufacturer max Position Value	0x1F00000
650BH		Serial Number	0xFFFFFFFF
2000H		Node ID (Plug Version)	0x1
2001H		Baudrate (Plug Version)	0x0

13. Store Parameters (Object 1010H)

The following parameters are stored via object 1010H:

Designation	**Object (Index/Subindex)**
TX PDO1 Identifier	1800/01
TX PDO2 Identifier	1802/01
EMCY Identifier	1014
SYNC Identifier	1005
NMT Life Guarding Identifier	100E
TX PDO 1 enabled	1800/01
TX PDO 2 enabled	1802/01
Node Number	100B
Baudrate	2001
TX PDO1 Transmission Type	1800/02
TX PDO2 Transmission Type	1802/02
TX PDO1 Inhibit Time	1800/03
TX PDO2 Inhibit Time	1800/03
Encoder Code Sequence	6000
Scaling Function Control enabled	6000
Units Function Control enabled	
Range Function Control enabled	
Preset Function Control enabled	
Measuring Units	6001
Measuring Range	6002
Scaled Resolution	
Preset Value Calculated (Offset)	6509
Preset Value	6003
Cyclic Timer	6200
Work Area Low Limit	6401
Work Area High Limit	6402
EEPROM Valid Flag	
EEPROM Checksum	

14. Literature

/1/ CiA Draft Standard Proposal 406 Version 2.0 Device Profile for Encoders

/2/ CiA Draft Standard 301 Version 3.0 CAL-based Communication Profile for Industrial Systems

/3/ CANopen Implementation Guidelines by G.Gruhler(Ed.) and Bernd Dreier Version 2.31
ESPRIT Project 22171 CANopen